

# Deprecated Workflow Generator

This page is **no longer maintained**. We have released a new *Workflow Generator* as part of the [WorkflowHub Project](#):

---

## Workflow Generator

To facilitate evaluation of workflow algorithms and systems on a range of workflow sizes, we have developed a set of synthetic workflow generators. These generators use the information gathered from actual executions of scientific workflows on the Grid as well as our understanding of the processes behind these workflows to generate realistic, synthetic workflows resembling those used by real world scientific applications.

The code used to generate all of the synthetic workflows below, and many others, is available from the GitHub repository. The java workflow generator sometimes generates negative task runtimes, so watch out for that.

## Simulator

WorkflowSim can be used to simulate the workflows generated by the Workflow Generator.

## Traces

Traces and execution logs from real workflows are available here: [here](#), [here](#), and [here](#). Data sets like these were used to parameterize the Workflow Generator.

## Synthetic Workflows

### Pegasus Workflows

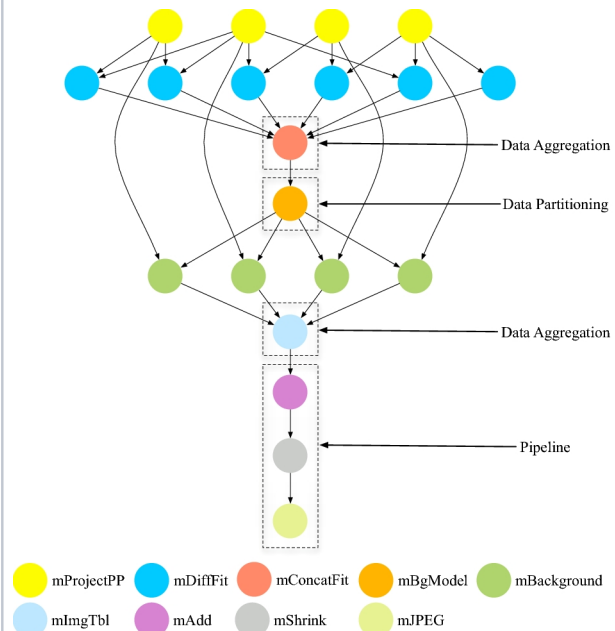
These workflows come from a [paper by Bharathi, et al. \[2\]](#). There is [another paper with more information about the workflows by Juve, et al. \[3\]](#).

[A large collection of DAXes similar to the ones listed below is available here](#). Note that it is about 375 MB.

Workflow Type	Example	DAX
---------------	---------	-----

**Montage**

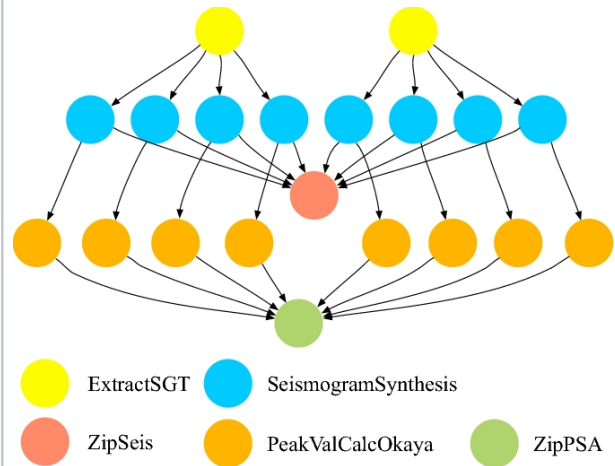
The Montage application created by NASA/IPAC stitches together multiple input images to create custom mosaics of the sky.



25 Node DAX  
50 Node DAX  
100 Node DAX  
1000 Node DAX

**CyberShake**

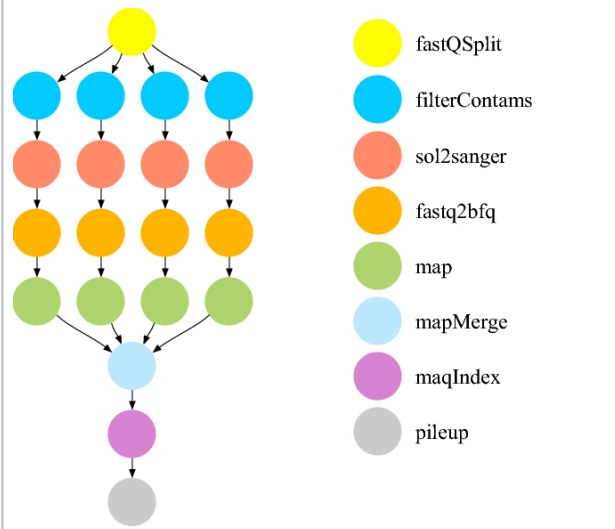
The CyberShake workflow is used by the Southern California Earthquake Center to characterize earthquake hazards in a region.



30 Node DAX  
50 Node DAX  
100 Node DAX  
1000 Node DAX

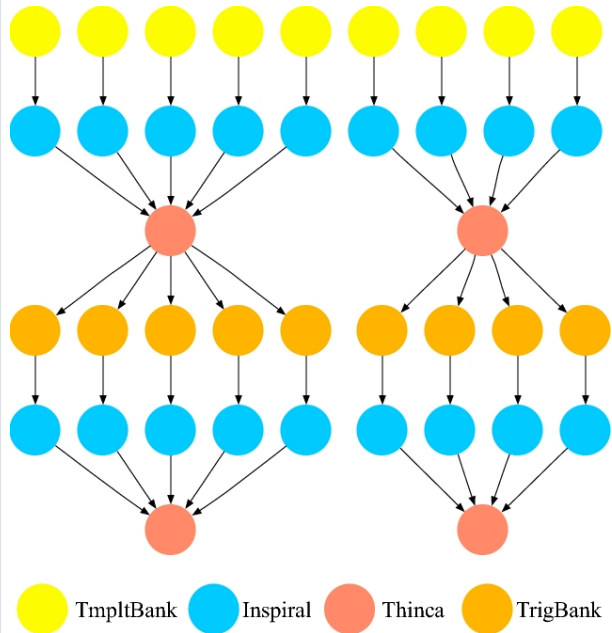
**Epigenomics**

The epigenomics workflow created by the USC Epigenome Center and the Pegasus Team is used to automate various operations in genome sequence processing.



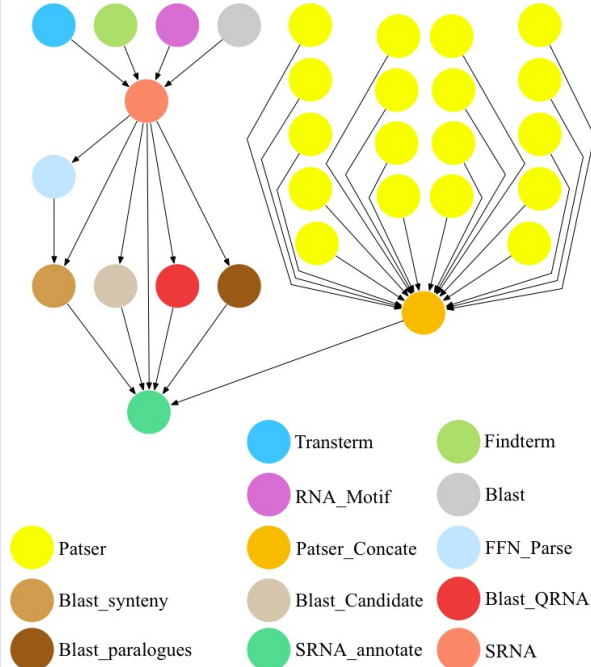
24 Node DAX  
46 Node DAX  
100 Node DAX  
997 Node DAX

**LIGO Inspiral Analysis**  
 LIGO's Inspiral Analysis workflow is used to generate and analyze gravitational waveforms from data collected during the coalescing of compact binary systems.



30 Node DAX  
 50 Node DAX  
 100 Node DAX  
 1000 Node DAX

**SIPHT**  
 The SIPHT workflow, from the bioinformatics project at Harvard, is used to automate the search for untranslated RNAs (sRNAs) for bacterial replicons in the NCBI database.



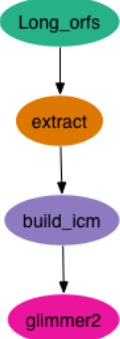
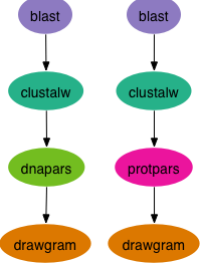
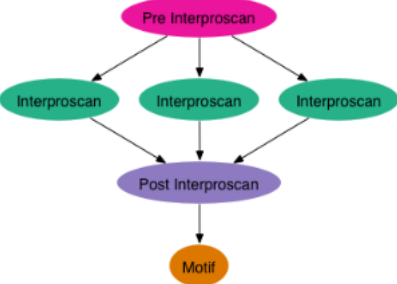

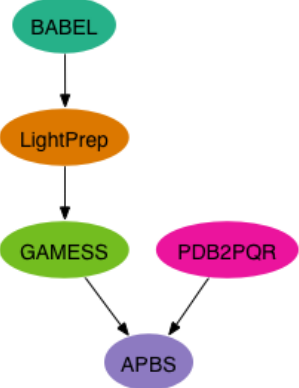
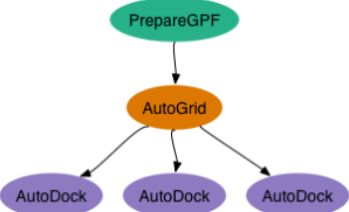
30 Node DAX  
 60 Node DAX  
 100 Node DAX  
 1000 Node DAX

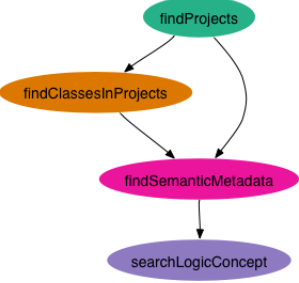
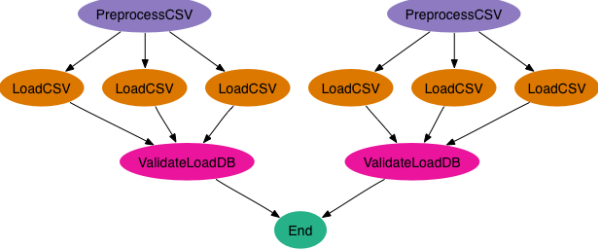
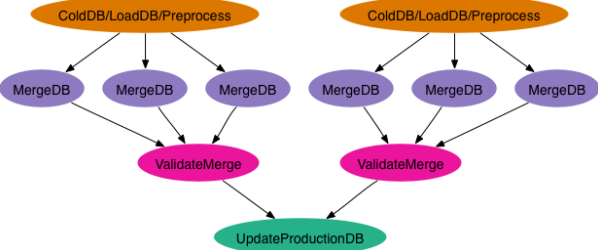
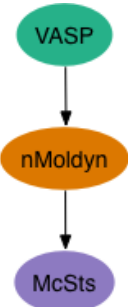
## Ramakrishnan and Gannon Workflows

These workflows come from a [report](#) by Ramakrishnan and Gannon [3].

Workflow Type	Figure in Report	Example	DAX
---------------	------------------	---------	-----

<b>LEAD Mesoscale Meteorology</b>	Figure 1	<pre> graph TD     TP[TerrainPreProcessor] --&gt; LB[LateralBoundaryInterpolator]     TP --&gt; 3D[3DInterpolator]     Wf[WrfStatic] --&gt; LB     LB --&gt; ARPS[ARPS2WRF]     3D --&gt; ARPS     ARPS --&gt; WRF[WRF]       </pre>	<a href="#">leadmm.xml</a>
<b>LEAD ARPS Data Analysis System</b>	Figure 2	<pre> graph TD     TP[TerrainPreProcessor] --&gt; LB[LateralBoundaryInterpolator]     TP --&gt; ADAS[ADASInterpolator]     Wf[WrfStatic] --&gt; LB     LB --&gt; ARPS[ARPS2WRF]     ADAS --&gt; ARPS     ARPS --&gt; WRF[WRF]       </pre>	<a href="#">leadadas.xml</a>
<b>LEAD Data Mining Workflow</b>	Figure 3	<pre> graph TD     SD[StormDetection] --&gt; RA[RemoveAttributes]     RA --&gt; SC[SpatialClustering]       </pre>	<a href="#">leaddm.xml</a>
<b>Storm Surge SCOOP Workflow</b>	Figure 4	<pre> graph TD     A1[Adcirc] --&gt; PP[PostProcessing]     A2[Adcirc] --&gt; PP     A3[Adcirc] --&gt; PP       </pre>	<a href="#">scoop_small.xml</a> <a href="#">scoop_medium.xml</a> <a href="#">scoop_large.xml</a>
<b>Floodplain Mapping</b>	Figure 5	<pre> graph TD     A1[Adcirc] --&gt; SWANON[SWAN Outer North]     A1 --&gt; SWANSO[SWAN Outer South]     A1 --&gt; SWANNI[SWAN Inner North]     A1 --&gt; SWANIS[SWAN Inner South]     A1 --&gt; A2[Adcirc]     WW[WaveWatchIII] --&gt; SWANON     WW --&gt; SWANSO     WW --&gt; SWANNI     WW --&gt; SWANIS     SWANON --&gt; A2     SWANSO --&gt; A2     SWANNI --&gt; A2     SWANIS --&gt; A2       </pre>	<a href="#">floodplain.xml</a>

Glimmer	Figure 6	 <pre> graph TD     A([Long_orfs]) --&gt; B([extract])     B --&gt; C([build_icm])     C --&gt; D([glimmer2]) </pre>	glimmer.xml
Gene2Life	Figure 7	 <pre> graph TD     A1([blast]) --&gt; B1([clustalw])     B1 --&gt; C1([dnapars])     C1 --&gt; D1([drawgram])     A2([blast]) --&gt; B2([clustalw])     B2 --&gt; C2([protpars])     C2 --&gt; D2([drawgram]) </pre>	gene2life.xml
Motif Network	Figure 8	 <pre> graph TD     A([Pre Interproscan]) --&gt; B1([Interproscan])     A --&gt; B2([Interproscan])     A --&gt; B3([Interproscan])     B1 --&gt; C([Post Interproscan])     B2 --&gt; C     B3 --&gt; C     C --&gt; D([Motif]) </pre>	motif_small.xml motif_medium.xml motif_large.xml
MEME-MAST	Figure 9	 <pre> graph TD     A([MEME]) --&gt; B([MAST]) </pre>	mememast.xml
Molecular Sciences	Figure 10	 <pre> graph TD     A([BABEL]) --&gt; B([LightPrep])     B --&gt; C([GAMESS])     B --&gt; D([PDB2PQR])     C --&gt; E([APBS])     D --&gt; E </pre>	molsci.xml
Avian Flu	Figure 11	 <pre> graph TD     A([PrepareGPF]) --&gt; B([AutoGrid])     B --&gt; C1([AutoDock])     B --&gt; C2([AutoDock])     B --&gt; C3([AutoDock]) </pre>	avianflu_small.xml avianflu_medium.xml avianflu_large.xml

caDSR	Figure 12	 <pre> graph TD     findProjects([findProjects]) --&gt; findClassesInProjects([findClassesInProjects])     findProjects --&gt; findSemanticMetadata([findSemanticMetadata])     findClassesInProjects --&gt; findSemanticMetadata     findSemanticMetadata --&gt; searchLogicConcept([searchLogicConcept])   </pre>	cadsr.xml
Pan-STARRS Load	Figure 13	 <pre> graph TD     PreprocessCSV1([PreprocessCSV]) --&gt; LoadCSV1([LoadCSV])     PreprocessCSV1 --&gt; LoadCSV2([LoadCSV])     PreprocessCSV1 --&gt; LoadCSV3([LoadCSV])     LoadCSV1 --&gt; ValidateLoadDB1([ValidateLoadDB])     LoadCSV2 --&gt; ValidateLoadDB1     LoadCSV3 --&gt; ValidateLoadDB1     PreprocessCSV2([PreprocessCSV]) --&gt; LoadCSV4([LoadCSV])     PreprocessCSV2 --&gt; LoadCSV5([LoadCSV])     PreprocessCSV2 --&gt; LoadCSV6([LoadCSV])     LoadCSV4 --&gt; ValidateLoadDB2([ValidateLoadDB])     LoadCSV5 --&gt; ValidateLoadDB2     LoadCSV6 --&gt; ValidateLoadDB2     ValidateLoadDB1 --&gt; End([End])     ValidateLoadDB2 --&gt; End   </pre>	psload_small.xml psload_medium.xml psload_large.xml
Pan-STARRS Merge	Figure 14	 <pre> graph TD     ColdDB1([ColdDB/LoadDB/Preprocess]) --&gt; MergeDB1([MergeDB])     ColdDB1 --&gt; MergeDB2([MergeDB])     ColdDB1 --&gt; MergeDB3([MergeDB])     MergeDB1 --&gt; ValidateMerge1([ValidateMerge])     MergeDB2 --&gt; ValidateMerge1     MergeDB3 --&gt; ValidateMerge1     ColdDB2([ColdDB/LoadDB/Preprocess]) --&gt; MergeDB4([MergeDB])     ColdDB2 --&gt; MergeDB5([MergeDB])     ColdDB2 --&gt; MergeDB6([MergeDB])     MergeDB4 --&gt; ValidateMerge2([ValidateMerge])     MergeDB5 --&gt; ValidateMerge2     MergeDB6 --&gt; ValidateMerge2     ValidateMerge1 --&gt; UpdateProductionDB([UpdateProductionDB])     ValidateMerge2 --&gt; UpdateProductionDB   </pre>	psmerge_small.xml psmerge_medium.xml psmerge_large.xml
McStas	Figure 15	 <pre> graph TD     VASP([VASP]) --&gt; nMoldyn([nMoldyn])     nMoldyn --&gt; McStas([McStas])   </pre>	mcstas.xml

[1] R. F. da Silva, W. Chen, G. Juve, K. Vahi, E. Deelman. Community Resources for Enabling Research in Distributed Scientific Workflows. 10th IEEE International Conference on e-Science (eScience 2014)

[2] S. Bharathi, A. Chervenak, E. Deelman, G. Mehta, M.-H. Su, and K. Vahi, "Characterization of Scientific Workflows", 3rd Workshop on Workflows in Support of Large Scale Science (WORKS 08), 2008.

[3] Gideon Juve, Ann Chervenak, Ewa Deelman, Shishir Bharathi, Gaurang Mehta, and Karan Vahi, "Characterizing and Profiling Scientific Workflows", *Future Generation Computer Systems*, 29:3, pp. 682–692, March 2013.

[4] L. Ramakrishnan and D. Gannon, "A Survey of Distributed Workflow Characteristics and Resource Requirements", Indiana University Technical Report TR671, 2008.